

Jobsheet
Inheritance dan Overriding

No	Kegiatan	Contoh Script	Latihan
		Java	
1	<i>Inheritance</i> <i>secara static</i>	<pre> public class burung{ String nama; String hidup; void terbang(){ System.out.println("Nama Burung : "+nama); System.out.println("Tempat Hidup : "+hidup); } } public class merpati extends burung { String makanan; String minuman; void konsumsi(){ System.out.println("Jenis makanan : "+makanan); System.out.println("jenis minuman : "+minuman); } } public class pewarisan { public static void main(String[] args){ burung BR = new burung(); merpati mp = new merpati(); System.out.println("KELAS BURUNG"); BR.nama = "Merpati"; BR.hidup = "Pohon"; BR.terbang(); mp.konsumsi(); } } </pre>	Buatlah program dengan menggunakan inheritance

2	Penggunaan <i>inheritance</i> dengan <i>modifier</i>	<pre> public class Bapak { int tangan; int kaki; public Bapak(){ this.tangan=2; this.kaki=2; } public void berdiri(){ System.out.println("Berdiri"); } public void berjalan(){ System.out.println("Berjalan"); } } public class Anak extends Bapak { public void bernyanyi(){ System.out.println("lalaa..alaa..."); } } public class BapakAnak { public static void main(String[] args) { Bapak bapak1 = new Bapak(); Anak anak1 = new Anak(); System.out.println("Jumlah Tangan = " + bapak1.tangan); bapak1.berjalan(); System.out.println("Jumlah kaki = " + anak1.kaki); anak1.berdiri(); } } </pre>	Buatlah program sederhana dengan menggunakan semua modifier dan inheritance
3	<i>Inheritance</i> secara statis	<pre> public class manusia { int umur; int get_umur(){ return umur; } } </pre>	Buatlah CV sederhana menggunakan <i>inheritance</i> .

	dengan <i>setter</i> <i>getter</i>	<pre> } class cv extends manusia{ public static void main(String[] args) { cv objek = new cv(); objek.set_umur(); System.out.println("umur = "+objek.get_umur()); } void set_umur(){ umur = 22; } } </pre>	Jumlah data sesuai NIM terakhir
4	<i>Inheritance</i> menggunakan lebih dari 2 <i>class</i>	<pre> public class bapak{ public void makan(){ System.out.println("makan"); } } public class anak extends bapak { public void menangis(){ System.out.println("menangis"); } } public class pewarisan { public static void main(String[] args){ bapak datasatu = new bapak(); anak datakedua = new anak(); datasatu.makan(); datakedua.menangis(); datakedua.makan(); } } </pre>	Buatlah konsep yang serupa menggunakan <i>inheritance</i> dengan data statis. Jumlah class sesuai dengan akhiran NIM
5	Menggunakan <i>inheritance</i>	<pre> class Anak{ Anak(){ System.out.println("Constructor of Parent"); } } </pre>	Buatlah program inheritance dengan menggunakan

	dan constructor	<pre> } } class BapakAnak extends Anak{ BapakAnak() { System.out.println("Constructor of Child"); } public static void main(String args[]){ new BapakAnak(); } } </pre>	constructor dimana objeknya sesuai dengan akhiran NIM
6	Membuat abstract class dan inheritance	<pre> abstract class Pulau { static final String NEGARA = "Indonesia"; public abstract String nama(); public String ambilNegara(){ return NEGARA; } } class Jawa extends Pulau { String namap; public String nama(){ namap = "Pulau Jawa"; return namap; } } class Kalimantan extends Pulau { String namap; public String nama(){ namap = "Pulau Kalimantan"; return namap; } } class JawaBarat extends Jawa { void namaProv(){ System.out.println("Ini Pulau Berada di "+" ambilNegara()); } } </pre>	Buatlah program yang menggunakan abstract class dan inheritance dengan class sesuai jumlah NIM

		<pre> System.out.println("Ini "+ nama()); System.out.println("Ini Provinsi Jawa Barat"); System.out.println("Jumlah Penduduk : 232342 Jiwa"); } } class KalimantanTimur extends Kalimantan { void namaProv(){ System.out.println("Ini Pulau Berada di "+ ambilNegara()); System.out.println("Ini "+ nama()); System.out.println("Ini Provinsi Kalimantan Timur"); System.out.println("Jumlah Penduduk : 32421 Jiwa"); } } class Utama { public static void main (String [] args){ JawaBarat JB = new JawaBarat(); KalimantanTimur KT = new KalimantanTimur(); JB.namaProv(); System.out.println(""); KT.namaProv(); } } </pre>	
7	Menggunakan inheritance dengan nilai dinamis	<pre> public class software { protected String code, namagame, lisensi; public software (){ } public software (String kode, String game, String lisen){ this.code = kode; this.namagame = game; this.lisensi = lisen; } public void setCode(String kode){ </pre>	Buatlah program inheritance secara dinamis, menggunakan statement control dan looping

		<pre> this.code = kode; } public String getCode(){ return code; } public void setNamagame(String game){ this.namagame = game; } public String getNamagame(){ return namagame; } public void setlisensi (String lisen){ this.lisensi = lisen; } public String getlisensi(){ return lisensi; } } </pre>	
		<pre> public class game extends software{ protected String jenis; protected String type; public game (String kode, String name, String lisen){ super(kode, name, lisen); } public game(String jns, String tipe){ this.jenis = jns; this.type = tipe; super.setCode(code); super.setNamagame(namagame); super.setlisensi(lisensi); } } </pre>	

		<pre> public String getCod(){ return super.getCode(); } public String getNama(){ return super.getNamagame(); } public String getlis (){ return super.getlisensi(); } public String getJenis(){ return jenis; } public String getType(){ return type; } } </pre>	
		<pre> import java.io.BufferedReader; import java.io.IOException; import java.io.InputStreamReader; public class main_2 { public static void main(String[] args) throws IOException{ BufferedReader input = new BufferedReader (new InputStreamReader(System.in)); String code; String nama; String jenis; String tipe; String lisen; int pilihan; game print = new game ("",""); game \$print = new game ("", "", ""); </pre>	

		<pre>System.out.println("+++++Game Pertama+++++"); System.out.println("-----"); System.out.println("Kode Game:"); print.code = code = input.readLine(); System.out.println("Nama :"); print.namagame = nama = input.readLine(); System.out.println("Jenis :"); print.jenis = jenis = input.readLine(); System.out.println("Type :"); print.type = tipe = input.readLine(); System.out.println("Lisensi :"); print.lisensi = lisen = input.readLine(); System.out.println(""); System.out.println("*****Game Kedua*****"); System.out.println("-----"); System.out.println("Kode Game:"); \$print.code = code = input.readLine(); System.out.println("Nama :"); \$print.namagame = nama = input.readLine(); System.out.println("Jenis :"); \$print.jenis = jenis = input.readLine(); System.out.println("Type :"); \$print.type = tipe = input.readLine(); System.out.println("Lisensi :"); \$print.lisensi = lisen = input.readLine(); System.out.println(""); System.out.println("Lihat Data"); System.out.println("1. Data Pertama\n" + "2. Data Kedua\n" + "0.Exit"); System.out.println("Masukan Pilihan : "); pilihan = Integer.parseInt(input.readLine()); while (pilihan != 0) { switch (pilihan) { case 1:</pre>	
--	--	---	--

		<pre> print.getNama()); System.out.println("Rincian Game" + System.out.println("====="); print.getCod()); System.out.println("Kode Game :" + -----"); print.getNama()); System.out.println("Nama Game :" + -----"); print.getJenis()); System.out.println("Jenis Game :" + -----"); print.getType()); System.out.println("Type Game :" + -----"); print.getlis()); System.out.println("Lisensi Game :" + -----"); break; case 2: \$print.getNama()); System.out.println("Rincian Game" + System.out.println("====="); \$print.getCod()); System.out.println("Kode Game : " + -----"); \$print.getNama()); System.out.println("Nama Game : " + -----"); \$print.getJenis()); System.out.println("Jenis Game : " + -----"); </pre>	
--	--	---	--

		<pre> System.out.println("----- -----"); \$print.getType(); System.out.println("Type Game : " + System.out.println("----- -----"); System.out.println("Lisensi Game : " + \$print.getlis()); System.out.println("----- -----"); break; default: } System.out.println("Lihat Data"); System.out.println("1. Data Pertama\n" + "2. Data Kedua\n" + "0.Exit"); System.out.println("Masukan Pilihan : "); pilihan = Integer.parseInt(input.readLine()); } } </pre>	
8	<i>Inheritance</i> menggunakan <i>overriding</i> dengan nilai statis	<pre> public class Animal { public String nama; public int umur; public void makan() { System.out.println("Sedang Makan"); } public void mengeluarkanSuara() { System.out.println("Sedang Bersuara"); } } public class Kucing extends Animal{ @Override public void mengeluarkanSuara() </pre>	Buatlah program menggunakan overriding dengan jumlah class dan overriding sesuai NIM terakhir

		<pre> { System.out.println("Meong"); } } } public class AnimalTest { public static void main(String[] args) { Animal animal1 = new Animal(); animal1.nama = "Binatang Abstrak"; animal1.mengeluarkanSuara(); Kucing kucing1 = new Kucing(); kucing1.nama = "Kitten"; kucing1.mengeluarkanSuara(); Animal animal2 = new Kucing(); animal2.nama = "Mew"; } } </pre>	
9	<i>Inheritance</i> dengan menggunakan <i>overriding</i> dan nilai dinamis	<pre> public class BangunDatar { float luas(){ System.out.println("Menghitung laus bangun datar"); return 0; } float keliling(){ System.out.println("Menghitung keliling bangun datar"); return 0; } } class Persegi extends BangunDatar { float sisi; @Override float luas(){ float luas = sisi * sisi; System.out.println("Luas Persegi: " + luas); } } </pre>	Buatlah program suatu rumus menggunakan inheritance dan overriding dengan nilai dinamis. Jumlah rumus sesuai dengan akhiran NIM

		<pre> return luas; } @Override float keliling(){ float keliling = 4 * sisi; System.out.println("Keliling Persegi: " + keliling); return keliling; } } </pre>	
		<pre> public class Lingkaran extends BangunDatar{ float r; @Override float luas(){ float luas = (float) (Math.PI * r * r); System.out.println("Luas lingkaran: " + luas); return luas; } @Override float keliling(){ float keliling = (float) (2 * Math.PI * r); System.out.println("Keliling Lingkaran: " + keliling); return keliling; } } </pre>	
		<pre> public class PersegiPanjang extends BangunDatar { float panjang; float lebar; @Override float luas(){ float luas = panjang * lebar; System.out.println("Luas Persegi Panjang:" + luas); return luas; } @Override float keliling(){ </pre>	

		<pre> float kll = 2*panjang + 2*lebar; System.out.println("Keliling Persegi Panjang: " + kll); return kll; } } </pre>	
		<pre> public class Segitiga extends BangunDatar { float alas; float tinggi; @Override float luas() { float luas = alas * tinggi; System.out.println("Luas Segitiga: " + luas); return luas; } } </pre>	
		<pre> public class Main { public static void main(String[] args) { BangunDatar mBangunDatar = new BangunDatar(); Persegi mPersegi = new Persegi(); mPersegi.sisi = 2; Lingkaran mLingkaran = new Lingkaran(); mLingkaran.r = 22; PersegiPanjang mPersegiPanjang = new PersegiPanjang(); mPersegiPanjang.panjang = 8; mPersegiPanjang.lebar = 4; Segitiga mSegitiga = new Segitiga(); mSegitiga.alas = 12; mSegitiga.tinggi = 8; mBangunDatar.luas(); mBangunDatar.keliling(); mPersegi.luas(); </pre>	

		<pre> mPersegi.keliling(); mLingkaran.luas(); mLingkaran.keliling(); mPersegiPanjang.luas(); mPersegiPanjang.keliling(); mSegitiga.luas(); mSegitiga.keliling(); } </pre>	
10	Penggunaan <i>abstract class</i> dan <i>abstract method</i> secara dinamis	<pre> public abstract class Hewan { abstract void setName(); abstract void setMakanan(); } public class Kelinci extends Hewan { public void setName(){ System.out.println("Nama hewan adalah \"KELINCI\""); } public void setMakanan(){ System.out.println("Makanan kelinci adalah \"WORTEL\""); } public void setWarna(){ System.out.println("Warna kelinci \"PUTIH\""); } public static void main(String[] args){ Kelinci k = new Kelinci(); k.setName(); k.setMakanan(); k.setWarna(); } } </pre>	Buatlah program dengan menggunakan abstract class dan abstract method dimana jumlah method sesuai dengan akhiran NIM
11	Penggunaan <i>overloading</i>	<pre> public class Overloading { public void jumlah (int a, int b){ System.out.println("Jumlah 2 angka =" + (a + b)); } } </pre>	Buatlah program menggunakan overloading

		<pre> public void jumlah (int a, int b, int c){ System.out.println("Jumlah 3 angka =" + (a + b + c)); } public void jumlah (double a, int b){ System.out.println("Jumlah 2 angka (double+int) = "+ (a + b)); } public void jumlah (int b, double a){ System.out.println("Jumlah 2 angka (int+double) = "+ (a + b)); } } </pre>	dengan nilai dinamis.dan jumlah method sesuai dengan akhiran NIM
		<pre> public class PenggunaanOverloading { public static void main(String[] args) { Overloading co = new Overloading(); co.jumlah(83,32); co.jumlah(34,454,432); co.jumlah(34.43,34); co.jumlah(28,33.23); } } </pre>	
12	Penggunaan <i>haybrid</i> <i>inheritance</i>	<pre> class C { public void disp() { System.out.println("C"); } } class A extends C { public void disp() { System.out.println("A"); } } class B extends C { public void disp() </pre>	Buatlah program dengan menggunakan hybrid inheritance dimana jumlah class sesuai dengan akhiran NIM dan bernilai dinamis

		<pre>{ System.out.println("B"); }</pre>	
		<pre>class D extends A { public void disp() { System.out.println("D"); } public static void main(String args[]){ D obj = new D(); obj.disp(); } }</pre>	